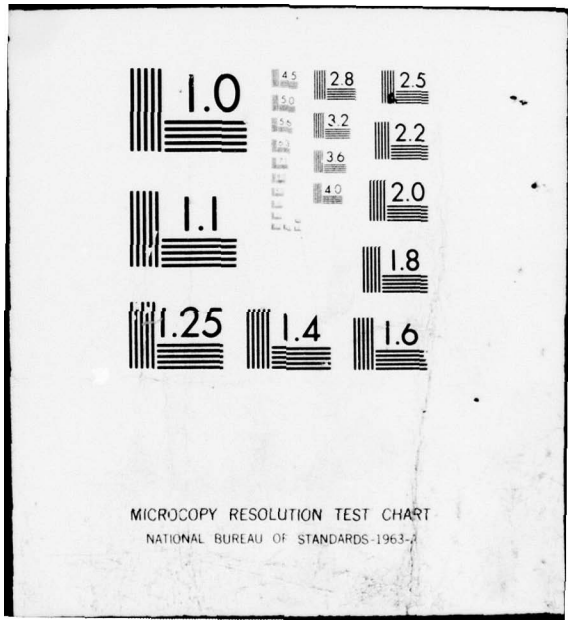
AD-A059 684 ARMY TANK-AUTOMOTIVE RESEARCH AND DEVELOPMENT COMMAND--ETC F/G 9/2
ENHANCED PLOTTING PACKAGE-USER'S MANUAL.(U)
JUL 78 F B HOOGTERP

UNCLASSIFIED TARADCOM-TR-12387 NL

| OF |
AD
A059684

END
DATE
FILMED
12-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TARADCOM R and D LABORATORY TECHNICAL REPORT

NO. 12387

ENHANCED PLOTTING PACKAGE— USER'S MANUAL

JULY 1978

by FRANCIS B. HOOGTERP

U.S. ARMY TANK-AUTOMOTIVE
RESEARCH AND DEVELOPMENT COMMAND
Warren, Michigan 48090

ENHANCED PLOTTING PACKAGE - USER'S MANUAL

JULY 1978

BY:

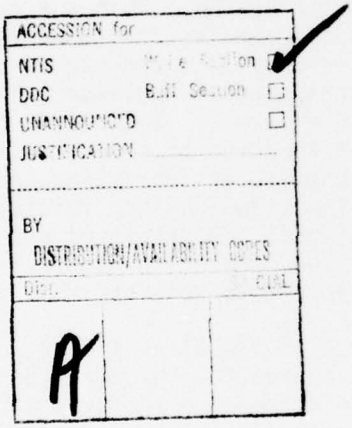
FRANCIS B. HOOGTERP

US ARMY TANK-AUTOMOTIVE
RESEARCH AND DEVELOPMENT COMMAND
WARREN, MI 48090

TABLE OF CONTENTS

ABSTRACT

This report describes a plotting package for use with the Tektronix 4014 graphics terminals on the Picatinny Arsenal CDC 6500/6600 time sharing system. The routines are used in conjunction with the TCS (Terminal Control System) provided on the CDC System, and are similar to several standard Calcomp plotting routines.



Introduction and Objective

In the course of his everyday duties, the research engineer is often called upon to use the digital computer as a tool to assist him in the solution of problems. One of the most difficult tasks when employing the computer is to utilize the large quantity of data that can be quickly generated. Through the use of interactive graphics, massive tables of results can often be reduced to a single plot. Graphics properly employed can not only assist in data interpretation, but can add instant documentation to a problem. The package of subroutines presented here is offered for general use in the hope that the overall engineering effort will be made more effective.

The routines are similar to CALCOMP plotting routines but allow more flexibility in defining the axes for graphs. This was done in order that graphs could be made on similar scales for comparison purposes. Before the potential user shies away from the use of graphics believing it too difficult and time consuming to implement, he should first look at the subroutine listed in Figure 2. This self-contained routine will plot a signal as a function of time on a Tektronix 4014 graphics terminal. This particular subroutine interactively requests the plot label and the signal units, initializes the graphics package and completes the plot. A sample of one such resulting plot is shown in Figure 3. The program in Figure 1 was used with the subroutine SIGPLT of Figure 2 to generate this plot.

It should be noted that the subroutine SIGPLT as listed, calls the routine AXISH to draw the axis. This causes the hardware character generator to be used for the axis annotations and the axis labels. These two calls could be replaced by calls to the routine AXISM. This would necessitate the change also in the size parameter from the integer variable ISIZE to an appropriate value of the real variable HLAB. These changes were made with HLAB = 0.15. The resulting plot from the modified SIGPLT is shown in Figure 4. In this case the axis annotations and labeling were performed using a software character generator.

Once a routine such as SIGPLT is written, it can be easily modified to perform similar tasks in other programs. I believe, in fact, if a computer user persistently makes use of a package such as this one it will soon become second nature to him. It will become as efficient (and much more desirable) to obtain plotted results as it would be to get nicely formatted output.

Plotting Procedures

One need not be familiar with TCS (the Terminal Control System available on the ARRADCOM system) to use the routines presented here. The only routine needed from TCS is the subroutine INITT. This routine must be called prior to the use of any other graphics routine. The call is always the same as shown in Figure 2. The only parameter in this call is the transmission rate and at present, all our graphics terminals transmit at 300 baud. This routine will initialize the graphics package and will erase the screen. The only other concession to TCS is that the PROGRAM statement card in the user's main program must contain the file names TAPE61 and TAPE62 in addition to any files the user desires. An example of this appears in the sample program of Figure 1.

The routines presented here are based on a common addressing scheme. The Tektronix 4014 screen is approximately 10 inches high and 14 inches wide. All screen position data for these subroutines will be given in inches measured from the lower left (the user's left) corner of the screen. The user will soon become adept at laying out his desired picture on a sheet of paper and picking off appropriate coordinates for input to the various routines.

The routines provided by this package include routines to move the cursor with the beam on or off, to rescale the total picture, to output variables or character strings, and to draw and grid either linear or logarithmic axes. Each routine that has a character drawing function associated with it (eg. the linear axis routine) has two separate forms. One form will use software generated characters that can be written with any size and orientation, but they require more time to be drawn. The other form of the routine uses hardware characters. The hardware characters can only be written horizonally (at an angle of 0 degrees) and are restricted to one of four standard sizes. Table 1 shows the index and relative size characteristics for each of the standard hardware characters. The use of hardware characters will significantly reduce picture generation time if very much labeling is used.

A procedure is provided to facilitate the incorporation of this graphics library with the user's compiled output. If the user has allowed his object file to go into the default LGO file, he will need only the following interactive statement to add the graphics routines required:

```
BEGIN, TEK, BR/HOOGTERP
```

The run file is then contained in the local file called TEK and may be cataloged or may be run by simply typing:

```
TEK
```

It is worthy to note that the usage of the BEGIN statement as described will obtain not only the required routines presented here, but also any routines required from TCS.

In review, if the user has his program requiring graphics in the editor, he may run it by simply typing the following commands:

```
RUN, FTN, N
BEGIN, TEK, BR/HOOGTERP
TEK
```

For the information of the more sophisticated user, the file names LGO and TEK used in the BEGIN/REVERT procedure are the first and second default parameters in the procedure. The file names LGO1 and TEK1 could be substituted for LGO and TEK respectively by using the following form of the BEGIN statement:

```
BEGIN, TEK, BR/HOOGTERP, LGO1, TEK1
```

Plotting Routine Descriptions

The following pages describe briefly each of the subroutines available and illustrates how they should be called. These subprograms exist in object form in the permanent library file MYTEKLIB with the ID = HOOGTERP.

All variables used in the call descriptions will conform to the FORTRAN defaults to determine their type. In other words, if a sample variable begins with a letter I through N inclusive, the corresponding user variable or constant must be of fixed point format (ie. an integer). If the variable in the description begins with any other letter, the corresponding user variable or constant must be of floating point format (ie. it must be a real value). The X distance refers to the horizonal distance from the left edge of the screen and the Y distance to the vertical distance from the lower edge of the screen.

PLOT

This routine is used to move the cursor from its present position to any desired new position. The move is performed in a straight line and can be accomplished with the beam either on or off. This routine can also be used to insure that the output buffer is dumped at a particular point in the program. The call for the routine is:

```
CALL PLOT (X, Y, IBEAM)
```

where

X - the x-coordinate to which the cursor is to be moved (all the coordinates are in inches from the lower left corner of the screen).

Y - the y-coordinate to which the cursor is to be moved.

IBEAM - this variable controls the beam status during the move and determines whether or not the buffer is to be transmitted to the terminal.

If IBEAM = 2 the beam is turned on during the move, drawing a visible line.
If IBEAM = 3 the beam is turned off during the move.
If IBEAM = 0 no movement is performed but the buffer contents are transmitted to the terminal.

FACT

This routine enables the user to enlarge or reduce the size of the entire plot. This can be accomplished by simply giving the scale factor by which all coordinates are to be multiplied. This subroutine need not be called. If it is not called, a default value of 1.0 is used for the scale factor. Generally one would not use this routine if hardware character writing routines were used since the size of the hardware characters would not be altered by a call to FACT. The call for this routine is of the form:

```
CALL FACT (SCALE)
```

where

SCALE - the ratio of the desired plot size to the normal plot size.

SYMB

This subroutine will place a string of characters at the specified location on the screen. The characters are software generated and may have any height and be drawn at any desired angle. The characters should be left justified in the array or variable. The routine may be called with the following statement:

```
CALL SYMB (X, Y, H, CHAR, ANGLE, NCHAR)
```

where

X, Y - the screen coordinates, in inches, of the lower left-hand corner (before character rotation) of the first character to be produced.

H - the height in inches, of the characters to be plotted.

CHAR - name of a variable or array containing the text to be plotted.

ANGLE - the angle, in degrees, measured counterclockwise from the X-axis, at which the annotation is to be plotted.

NCHAR - the number of characters in the array to be plotted.

HSYMB

This routine is similar in function to the subroutine SYMB. This routine, however, uses hardware characters instead of the software characters generated by SYMB. As a result HSYMB can only print characters horizonally (equivalent to setting ANGLE = 0 in SYMB). This routine is also limited to a selection of four character sizes. The exact dimensions of each size are given in Table 1. As with SYMB the text should be left justified. The call for HSYMB is:

```
CALL HSYMB (X, Y, ISIZE, CHAR, NCHAR)
```

where

X, Y - the screen coordinates (in inches) of the lower left corner at the first character to be printed.

ISIZE - the size of the hardware character to be used (1-4 where 1 is the largest size).

CHAR - variable or array containing text to be printed.

NCHAR - the number of characters to be printed.

NOTE: This routine has one less parameter that its software character generating counterpart, SYMB.

NUMB

This routine takes a floating point variable or constant, converts it to a character string and uses SYMB to place its value on the screen at the specified location. The software character capability of SYMB is extended through this routine to the drawing of numbers. The call for NUMB is as follows:

```
CALL NUMB (X, Y, H, VAL, ANGLE, N)
```

where

X, Y - the x and y screen coordinates (in inches) of the lower left corner of the plotted number.

H - the height in inches of the number to be drawn.

VAL - the floating point number to be drawn.

ANGLE - the angle, in degrees, measured counterclockwise from the x-axis, at which the number is to be plotted.

N - If N > 0, it specifies the number of digits to the right of the decimal point to be drawn.
If N=0, no digits will be drawn to the right of the decimal point.

If N= -1, the decimal point will also be suppressed.

HNUMB

This routine performs similarly to NUMB with the exception that the hardware character routine HSYMB is called to actually place the number on the screen. The size and angle restrictions of HSYMB apply here also. The call for HNUMB is:

```
CALL HNUMB (X, Y, ISIZE, VAL, N)
```

where

X, Y - the screen coordinates (in inches) of the lower left corner of the number printed.

ISIZE - the hardware character size (1-4 where 1 means the largest size).

VAL - the floating point variable or constant to be printed.

N - the number of digits desired to the right of the decimal point (for N<0 the decimal point will also be suppressed).

NOTE: The angle parameter included in NUMB is not available in this routine.

AXISM

This subroutine allows the user to have one axis of a graph drawn and labeled with a single call. The axis can be drawn at any angle and software characters are used for labeling. Two sizes of tic marks are generated with the smaller one appearing between axis annotations and a larger one placed at each annotated point. This routine is called by:

```
CALL AXISM (X,Y,ALAB,NLAB,ALEN,ANG,AMIN,SCALE,HLAB,NTICM,
           LREP,IDP)
```

where

X - x coordinate of axis origin.

Y - y coordinate of axis origin.

ALAB - array or variable containing axis label.

NLAB - number of characters in axis label.
If NLAB < 0 the labeling will be done on the clockwise side of the axis.
If NLAB > 0 the labeling is done on the counterclockwise side.

ALEN - length of axis (inches).

ANG - angle to axis measured counterclockwise from the screen x-axis in degrees.

AMIN - minimum value on axis.

SCALE - scale factor on axis (units/inch)

HLAB - height (in inches) of axis label and axis annotation (HLAB = 0, no labeling).

NTICM - number of tic mark intervals.

LREP - label repeat cycle (number of tic mark intervals per axis annotation).

IDP - number of digits after decimal point, in axis annotation.

AXISH

This routine is similar to AXISM in that it allows the user to draw and label one axis of a graph with a single call. This routine, however, utilizes the hardware characters. As a result the time to draw and label the axis is lessened considerably at the cost of some flexibility. The call is very similar to the call of AXISM with only the character size parameter changing. Also, the angle entered is rounded to the nearest multiple of 90 degrees and either a vertical or a horizontal axis is then drawn. The label for a vertical axis will run vertically from top to bottom. This subroutine also generates two sizes of tic marks. Smaller ones appear between axis annotations and larger ones are placed at each annotated point. The call for this routine is:

```
CALL AXISH(X,Y,ALAB,NLAB,ALEN,ANG,AMIN,SCALE,ISIZE,NTICM,
          LREP,IDP)
```

where

X, Y - coordinates of axis origin.

ALAB - array or variable containing axis label.

NLAB - number of characters in axis label.
if NLAB<0 the labeling will be on clockwise side of axis.
if NLAB>0 the labeling will be on counterclockwise side.

ALEN - length of axis (inches).

ANG - angle to axis measured counterclockwise from the screen x-axis in degrees.

AMIN - minimum value on axis.

SCALE - scale factor on axis (units/inch).

ISIZE - the size of the hardware characters to be used (1-4 where 1 is the largest size).

NTICM - number of tic mark intervals.

LREP - label repeat cycle (number of tic mark intervals per axis annotation).

IDP - number of digits to the right of the decimal point in axis annotations.

LOGAX

This routine will draw an axis with a logarithmic scale, using the software character generating capability. The user is allowed to select the angle at which the axis is to be drawn but the labeling size is selected for him. The calling statement is of the form:

```
CALL LOGAX (X,Y,ALAB,LCNT,ALEN,ANG,EXPMX,SCALE)
```

where

X - x coordinate of axis origin.

Y - y coordinate of axis origin.

ALAB - array containing the axis label.

LCNT - number of characters in axis label (sign convention same as that in AXISM and AXISH).

ALEN - length of axis (inches).

ANG - angle to axis measured counterclockwise from the screen X axis in degrees.

EXPMX - maximum exponent value (maximum power of 10).

SCALE - number of inches per cycle along the axis.

LOGAXH

This subroutine draws either a vertical or a horizontal axis with a logarithmic scale. The lettering is done using the hardware character generator on the Tektronix 4014. As in LOGAX the character size for labeling is not available for selection by the user. The call for this routine is:

```
CALL LOGAXH (X,Y,ALAB,LCNT,ALEN,ANG,EXPMX,SCALE)
```

where

X, Y - the origin of the axis (in inches).

ALAB - the axis label.

LCNT - the number of characters in the label (sign convention same as that in AXISH and AXISM).

ALEN - length of axis (in inches).

ANG - angle of axis (0 for horizonal; otherwise vertical).

EXPMX - maximum exponent value (maximum power of 10).

SCALE - scale factor for axis (inches per cycle).

LOGRD

This subroutine can be used to draw grid lines on a graph. The grid lines can have either a linear or a logarithmic spacing. One call to LOGRD will produce one set of grid lines. For example, one call is needed for the horizonal grid lines and another for the vertical grid lines. The routine is used with the following call:

```
CALL LOGRD (XORG,YORG,GRLEN,AXLEN,SCAL,ANG)
```

where

XORG,YORG - the origin coordinates for the corresponding axis.

GRLEN - length of grid lines (>0 for counterclockwise from corresponding axis; <0 for clockwise).

AXLEN - length of corresponding axis in inches.

SCAL - if SCAL >0 then this is the scale factor for corresponding logarithmic axis (inches/cycles).
if SCAL <0 then the absolute value of SCAL is used as the spacing for a linear grid (in inches).

ANG - angle of corresponding axis (degrees counterclockwise from X-axis).

An example might prove beneficial in this case. Let us presume we have just drawn linear x and y axes of length XLEN and YLEN respectively and assume also that they are parallel to the respective screen x and y axes. Then the horizonal grid lines (those corresponding to the vertical y axis) can be generated with the following call:

```
CALL LOGRD(XORG,YORG,-XLEN,YLEN,-DELY,90.)
```

where DELY is the desired spacing of the grid lines in inches.

The call for the vertical grid lines (those corresponding to the horizonal x axis) is as follows:

```
CALL LOGRD(XORG,YORG,YLEN,XLEN,-DELX,0.)
```

where DELX is the desired linear spacing of the vertical grid lines in inches.

PREP

This subroutine is used by NUMB and HNUMB. This routine does not perform graphics but is included just the same. The subroutine converts a floating point number into a left justified character string. This string of characters can then be output without formatting.

The number of digits to the right of the decimal point is specified by the user and the decimal point can be suppressed if desired. The length of the character string is returned along with the string itself. The call for this routine is:

```
CALL PREP (VAL, N, STRG, LEN)
```

where

VAL - the floating point number to be converted to a character string.

N - the number of digits desired to the right of the decimal place.
If N < 0, the decimal point is not placed in the character string.

STRG - output string of characters to represent number.

LEN - length of output string (number of characters).

```
SAMPLE

      PROGRAM SAMPLE(INPUT,OUTPUT,TAPE61,TAPE62)
      DIMENSION TIME(100),SIGNAL(100)
      CALL CONNEC(5LINPUT)
      CALL CONNEC(6LOUTPUT)
      NPOINT=100
C
C    CONSTRUCT SAMPLE SIGNAL
C
      DO 10 I=1,NPOINT
      TIME(I)= (I-1)/10.
      SIGNAL(I)= SIN(0.628*TIME(I) )
   10 CONTINUE
      CALL SIGPLT(-1.,1.,TIME,SIGNAL,NPOINT)
      STOP
      END
```

Figure 1 Sample Main Program

```
SIGPLT


C
C    THIS SUBROUTINE PLOTS THE SIGNAL SPECIFIED AS A
C    FUNCTION OF TIME.
C
C
      SUBROUTINE SIGPLT(AMIN,AMAX,TIME,SIGNAL,NPOINT)
      DIMENSION TIME(1),SIGNAL(1),LABEL(5)
      DATA LABT,LUNITS /10HTIME--SEC ,10HAMPLITUDE /
      DATA XORG,YORG /1.5,2.0/
C
C    USER ENTERS LABEL AND SIGNAL AXIS LIMITS
C
      PRINT 101
  101 FORMAT (* ENTER PLOT LABEL *)
      READ 102,LABEL
  102 FORMAT (5A10)
      PRINT 100,AMIN,AMAX
  100 FORMAT (* MINIMUM VALUE =*,F10.4,*   MAXIMUM VALUE =*,F10.4,
     1    *   AXIS MIN AND MAX + *,/)
      READ *,YMIN,YMAX
      TMAX= INT(TIME(NPOINT) + 3.99)  / 4   * 4
      SCALT=TMAX/12. + 0.001
      SCALY=(YMAX-YMIN) / 8.
C
C    INITIALIZE GRAPHICS PACKAGE, PLOT TIME AXIS, THEN Y AXIS.
C
      CALL INITT(300)
      CALL AXISH(1.5,6.,LABT,-9,12.,0.,0.,SCALT,2,8,2,1)
      CALL AXISH(1.5,2.,LUNITS,10,8.,90.,YMIN,SCALY,2,8,2,1)
      T=TIME(1)/SCALT  + XORG
      Y= (SIGNAL(1)-YMIN) / SCALY  + YORG
C
C    PLOT SIGNAL AS FUNCTION OF TIME
C
      CALL PLOT(T,Y,3)
      DO 40 I=2,NPOINT
      T= TIME(I) / SCALT  + XORG
      Y= (SIGNAL(I) - YMIN) / SCALY  + YORG
      CALL PLOT(T,Y,2)
   40 CONTINUE
      CALL HSYMB(4.,1.,2,LABEL,50)
C
C    TRANSMIT TOTAL PICTURE TO SCREEN AND PLACE TERMINAL
C    IN WAIT STATE.  THIS ALLOWS A CLEAN COPY TO BE MADE.
C    THE PROGRAM WILL CONTINUE WHEN ANY NUMBER IS ENTERED.
C
      CALL PLOT(0.,0.,-3)
      READ *,DUM
      RETURN
      END
```

Figure 2 Subroutine to Plot Time Signal

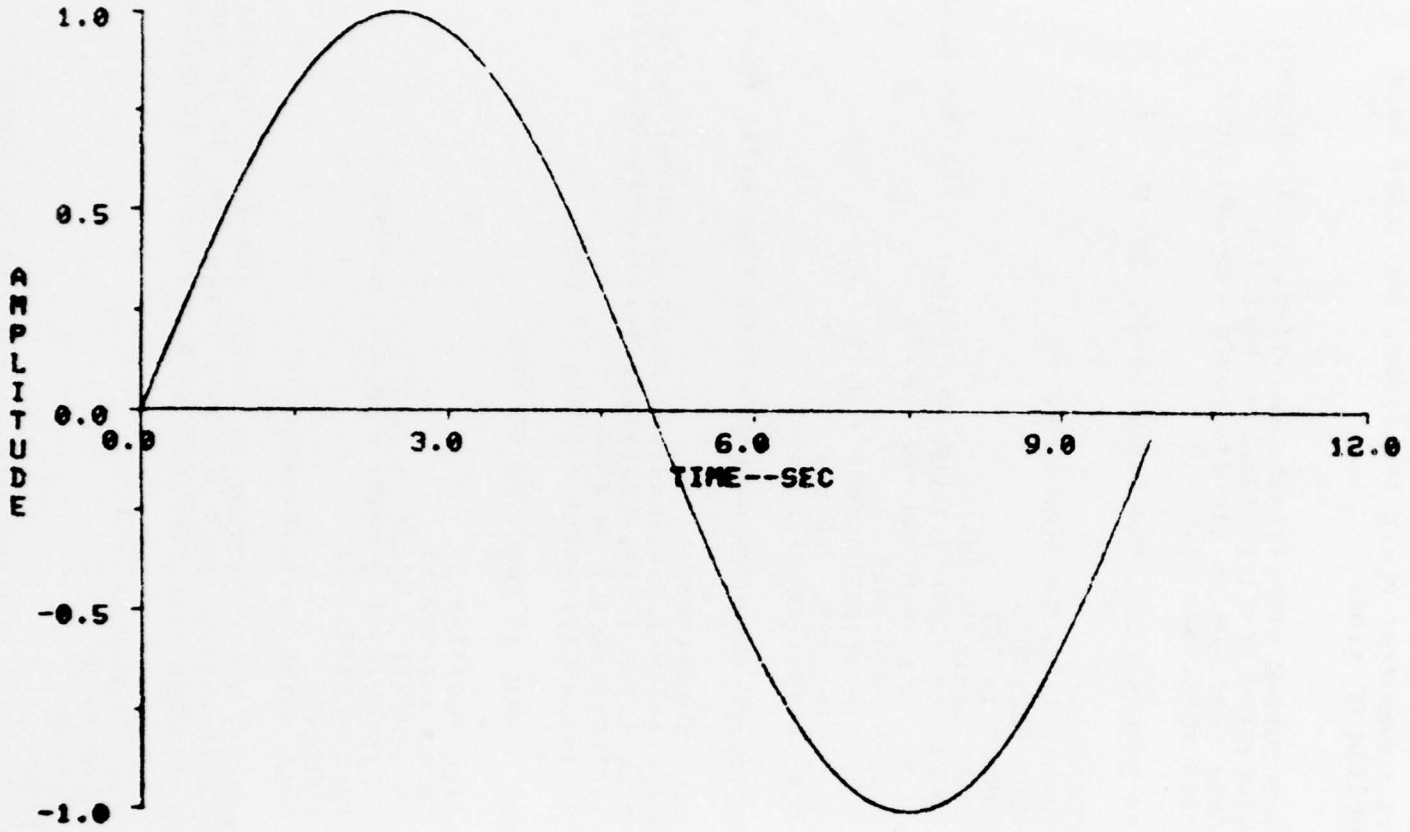


Figure 3 SAMPLE PLOT OF SINE WAVE

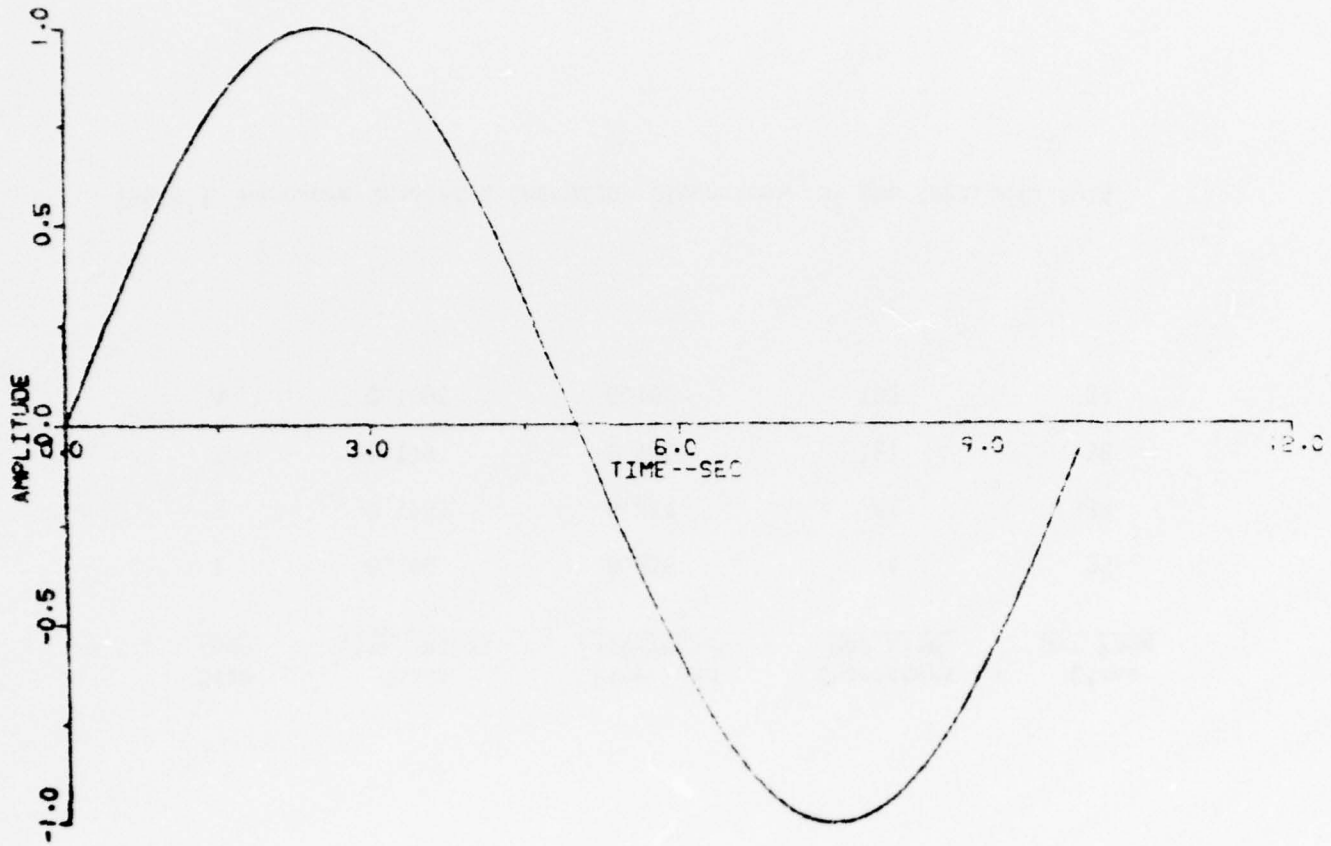


Figure 4 **SAMPLE PLOT USING SOFTWARE GENERATED CHARACTERS**

Size Code	Space Size-inches	Line Feed Size-inches	Characters per Line	Lines per Page
1	0.196	0.315	74	35
2	0.1785	0.287	81	38
3	0.119	0.1855	121	58
4	0.1085	0.168	133	64

Table 1 Hardware Generated Character Dimensions for the Tektronix 4014

DISTRIBUTION LIST

	No. of Copies
Commander US Army Tank-Automotive Research & Development Command	
ATTN: Applied Research Function, DRDTA-ZSA	20
Survival Technology Function, DRDTA-ZSC	5
Technical Library, DRDTA-UL	3
Warren, MI 48090	
Commander Defense Documentation Center Cameron Station Alexandria, VA 22314	12

(14) TARADCOM-TR-12387

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 12387
2. GOVT ACCESSION NO. (9) Technical rept.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) Enhanced Plotting Package-User's Manual
5. TYPE OF REPORT & PERIOD COVERED
6. PERFORMING ORG. REPORT NUMBER: 12387

7. AUTHOR(s): (10) FRANCIS B. HOOGTERP
8. CONTRACT OR GRANT NUMBER(s): (12) 27p.

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
US Army Tk-Autmv Rsch & Dev Command
DRDTA-ZSA
Warren, MI 48090

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS:
US Army Tk-Autmv Rsch & Dev Command
DRDTA-UL
Warren, MI 48090

12. REPORT DATE: July 1978
13. NUMBER OF PAGES: 22

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Software
Graphics Software
Plotting Software

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a plotting package for use with the Tektronix 4014 graphics terminals on the Picatinny Arsenal CDC 6500/6600 time sharing system. The routines are used in conjunction with the TCS (Terminal Control System) provided on the CDC System, and are similar to several standard Calcomp plotting routines.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 882